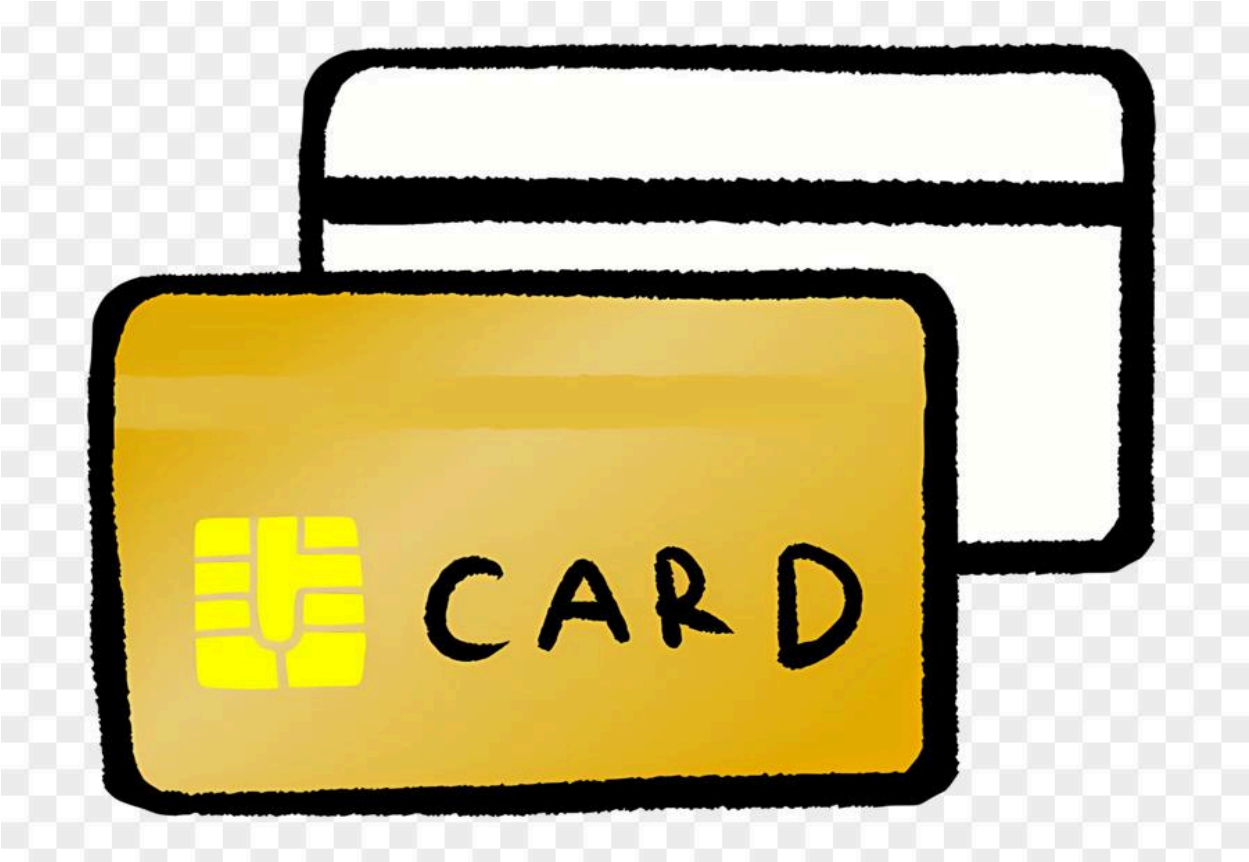


CREDIT CARD APPROVAL PREDICTION

02.02.2024

—

SAIMANOJ KANDUKURI
Data Scientist
EMAIL
Linkedin

- *projectlink*

1. Problem Statement

A bank's credit card department is one of the top adopters of data science. A top focus for the bank has always been acquiring new credit card customers. Giving out credit cards without doing proper research or evaluating applicants' creditworthiness is quite risky. The credit card department has been using a data-driven system for credit assessment called Credit Scoring for many years, and the model is known as an application scorecard. A credit card application's cutoff value is determined using the application scorecard, which also aids in estimating the applicant's level of risk. This decision is made based on strategic priority at a given time.

Customers must fill out a form, either physically or online, to apply for a credit card. The application data is used to evaluate the applicant's creditworthiness. The decision is made using the application data in addition to the Credit Bureau Score, such as the FICO Score in the US or the CIBIL Score in India, and other internal information on the applicants. Additionally, the banks are rapidly taking a lot of outside data into account to enhance the caliber of credit judgements.

Challenges:

Risk Mitigation: The existing credit scoring model might not be capturing all relevant factors, leading to potential risks associated with approving credit cards to undeserving applicants.

Data Integration: With the rapid influx of external data, integrating and leveraging diverse data sources for credit judgments has become a complex task.

Enhancing Decision Dynamics: The strategic priorities of the bank evolve over time, necessitating a dynamic credit approval system that adapts to changing circumstances.

Why these proposal important in today's world ?

In today's world, where data-driven decision-making is integral to business success, the proposal for implementing an advanced credit card approval system holds significant importance. Predicting a good client is crucial for a bank for several reasons:

Risk Mitigation: Identifying creditworthy clients mitigates default risk, ensuring financial stability.

Efficiency: Accurate credit predictions optimize resource allocation, reducing follow-up efforts on risky accounts.

Customer Satisfaction: A robust credit approval system enhances customer experience, fostering positive relationships.

Competitive Advantage: Predictive modeling sets banks apart, attracting creditworthy clients and enhancing reputation.

Compliance: Advanced credit systems align with regulatory guidelines, reducing legal and regulatory risks.

Financial Inclusion: Accurate credit assessments contribute to economic inclusivity, supporting individuals with untapped financial potential.

The impact on the banking sector is multi-faceted:

Enhanced Decision-Making:

- Incorporates machine learning for accurate predictions.
- Adapts to changing economic conditions.

Operational Efficiency:

- Streamlines credit approval processes.
- Particularly beneficial for large-scale operations.

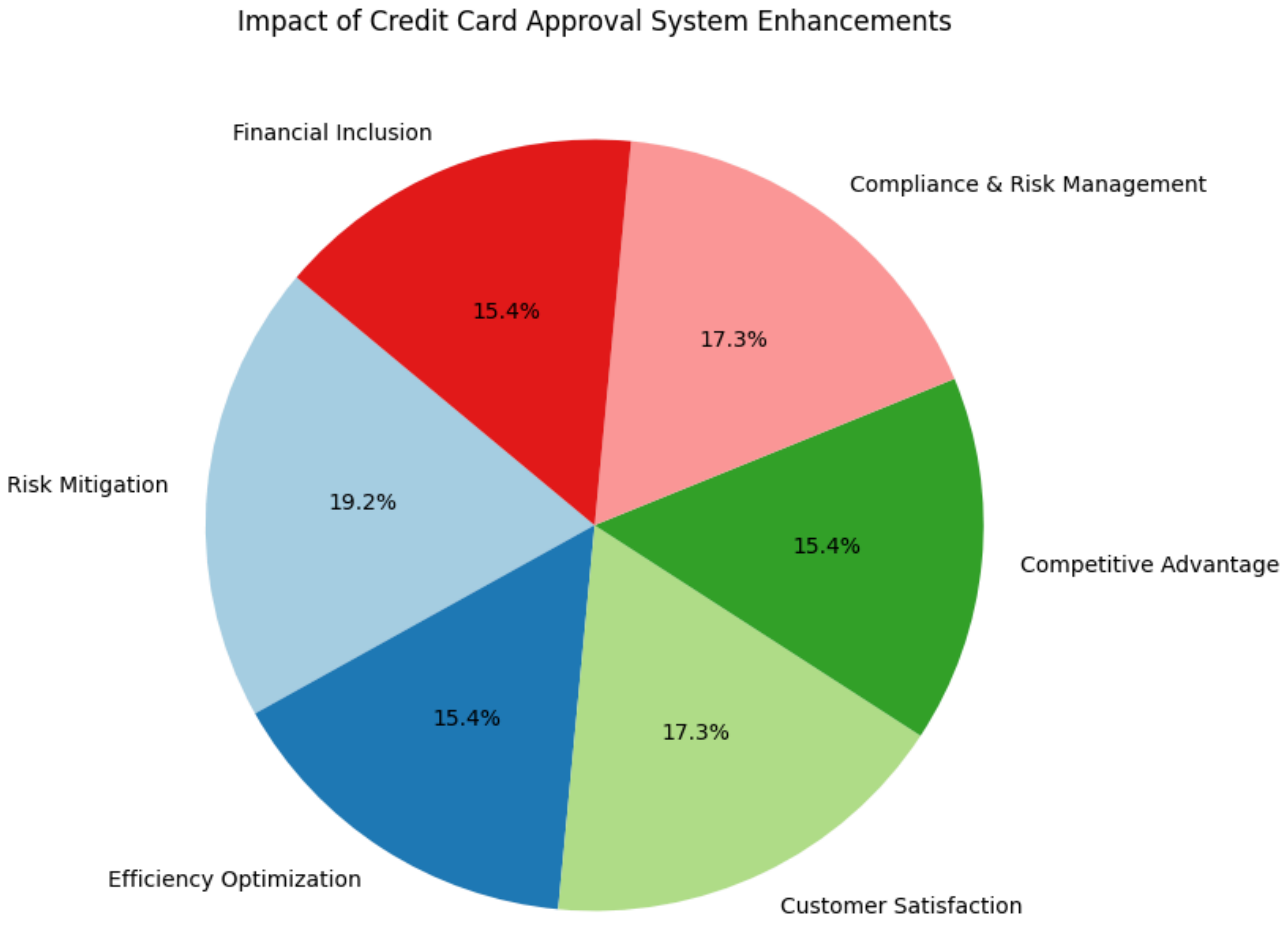
Reduced Non-Performing Loans (NPLs):

- Identifies creditworthy clients accurately.
- Mitigates the risk of Non-Performing Loans.

Adaptability to Market Dynamics:

- Machine learning models adjust to market changes.
- Ensures system effectiveness in evolving conditions.

In summary, the proposal revolutionizes credit card approval, boosting accuracy, efficiency, and alignment with banking sector needs. The impact extends to risk management, customer satisfaction, and overall competitiveness in the dynamic financial landscape.



2. Project Overview

2.1 Project Goal

- Primary Goal:
    - Develop an advanced credit card approval system.
    - Utilize data science and machine learning for precise credit assessments.
    - Minimize risks in credit card approvals.
    - Maximize acquisition of creditworthy customers.
- Secondary Goal:
    - Extract insights from the dataset.
    - Inform further decision-making for the Credit card company.

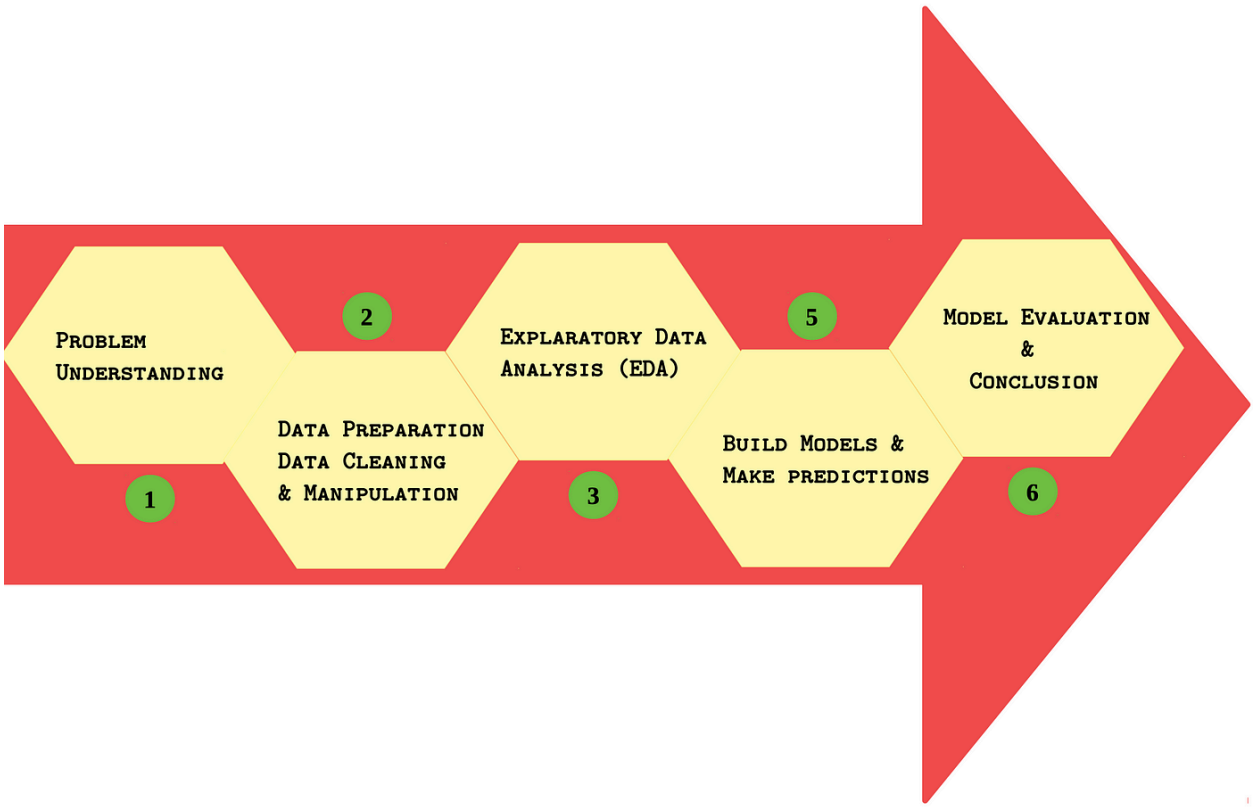
2.2 Objectives

- Develop a predictive model that considers a wide array of factors, including traditional credit factors , internal information, and external data sources.
- Implement a dynamic credit approval system that adapts to the strategic priorities of the bank.
- Enhance the overall efficiency and accuracy of credit assessments, leading to improved risk management.

3. Proposed Solution

3.1 Approach

The proposed solution involves the implementation of an advanced credit card approval system using machine learning techniques, with a specific emphasis on the XGBoost model with Recursive Feature Elimination (XGBoost RFE). The model aims to provide a more accurate assessment of creditworthiness by considering a comprehensive set of features.



3.2 Methodology

3.2.1.Data Collection:

Gather historical credit card application data Credit_card.csv',

Credit_card_label.csv.

3.2.2.Data Understanding:

- Prioritize understanding data features as the foundational step.
- Distinguish between numerical and categorical data.
- Identify nominal and ordinal categories, as well as continuous and discrete variables.

3.2.3.DATA CLEANING

Trimming and Duplicate Removal:

- Remove trailing spaces and identify duplicate records.
- Eliminate duplicate entries to ensure data accuracy.

Column Handling:

- Remove empty columns for streamlined datasets.
- Rename features for clarity and consistency.

Missing Values:

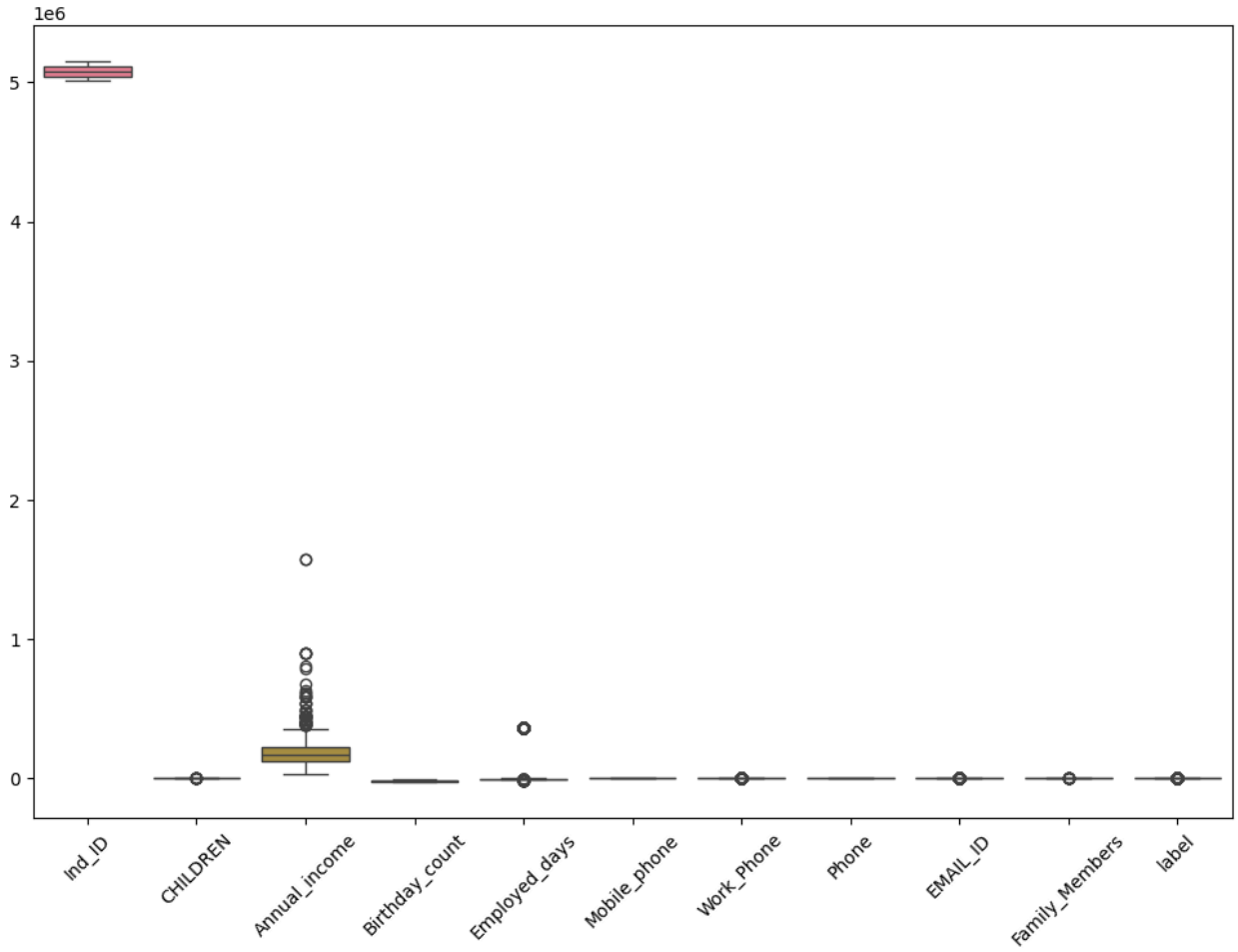
- Identify and handle missing values using Python libraries.
- Utilize imputation techniques, filling numerical values with measures of central tendency (MCT).
  - Mean for variables with no outliers.
  - Median if outliers are present.
  - Mode for other variables.

Handling Outliers:

- Identify outliers through visual inspection (box plots, scatter plots, histograms) and statistical methods (Z-scores, IQR).
- Use Interquartile Range (IQR) for robust outlier identification.
  - IQR is resistant to skewed distributions and less influenced by extreme values.
- Treat outliers through capping, setting a threshold to manage extreme values.

Objective:

Ensure a clean, standardized dataset by addressing duplicates, empty columns, missing values, and outliers, laying a solid foundation for subsequent analysis and modeling.



3.2.4.Data Exploration and Cleaning : EDA

Exploring the dataset to understand its structure and identifying patterns or anomalies. Handle missing values, outliers, and inconsistencies. This step often involves data visualization and statistical analysis

The features from dataset are

Ind_ID, GENDER, Car_Owner Propert_Owner, CHILDREN,Annual_income Type_Income, EDUCATION, Marital_status, Housing_type , Birthday_count Employed_days , Mobile_phone , Work_Phone , Phone , EMAIL_ID Type_Occupation , Family_Members , label

Hypothesis testing is employed to determine the statistical relationship between different variables, evaluating whether observed associations are random or statistically significant.

Setting Hypotheses:

- Formulate null and alternate hypotheses to establish the basis for testing.

P-Value Comparison:

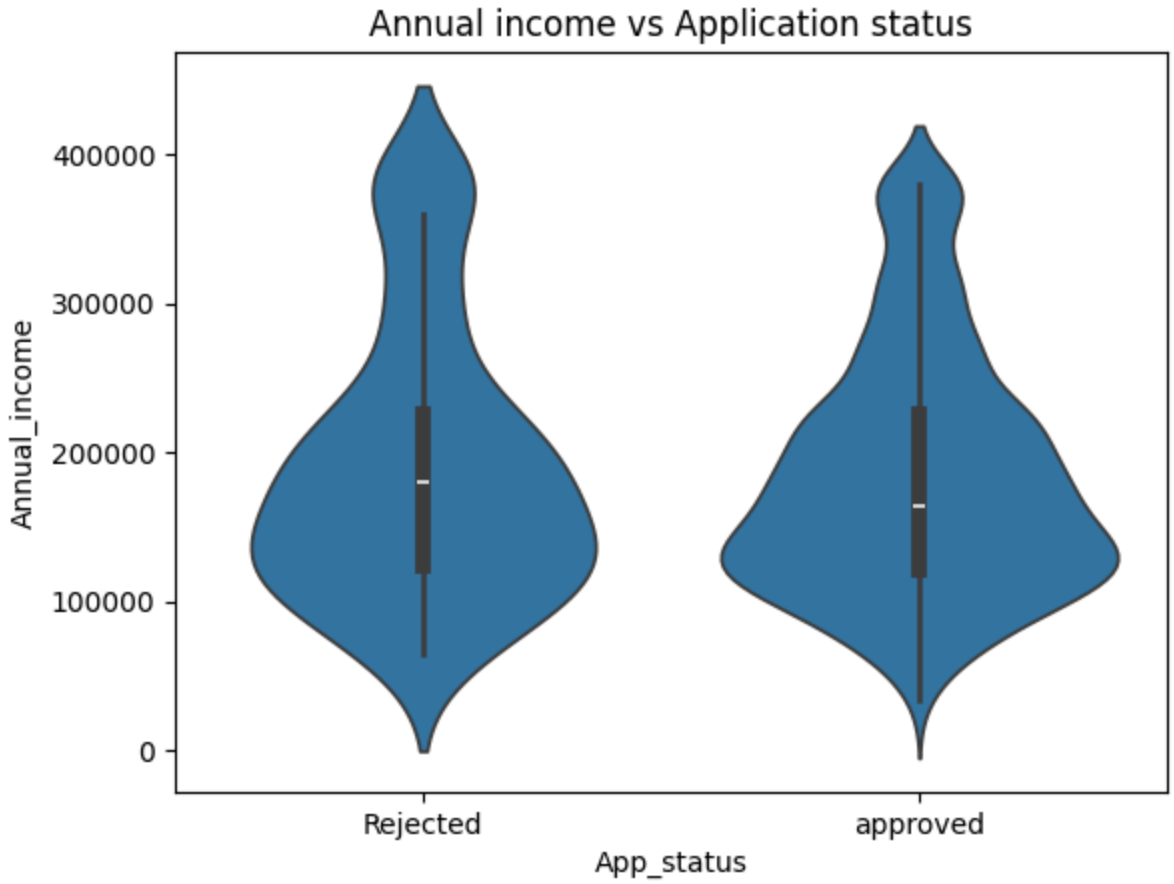
- Compare the calculated P-value with the chosen level of significance.
- If P < level of significance, reject the null hypothesis.
- If P > level of significance, fail to reject the null hypothesis, considering the alternate hypothesis.

Project Observations:

- No statistical relation is observed between application status and gender, car ownership status, Ind_ID, marital status, and education.
- Statistical relations are identified between family members and children, phone and work phone.
- New features, like age and working experience, exhibit correlation between working days and birthday count, as inferred from the heatmap correlation graph.
- Significant correlation exists between annual income and application status, as well as between type of income and application status.

Objective: Provide a clear understanding of the hypothesis testing process and highlight specific observations from the project, emphasizing significant and non-significant relationships among variables.


CHILDREN
Annual_income
Birthday_count
Employed_days
Family_Members
Absolute_Employed_days
Years_of_Employment
age
1 0.067 0.32 -0.19 0.87 -0.05 -0.05 -0.32
0.067 1 0.11 -0.15 0.052 -0.045 -0.045 -0.11
0.32 0.11 1 -0.34 0.29 -0.37 -0.37 -1
-0.19 -0.15 -0.34 1 -0.21 -0.43 -0.43 0.34
0.87 0.052 0.29 -0.21 1 -0.039 -0.039 -0.29
-0.05 -0.045 -0.37 -0.43 -0.039 1 1 0.37
-0.05 -0.045 -0.37 -0.43 -0.039 1 1 0.37
-0.32 -0.11 -1 0.34 -0.29 0.37 0.37 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

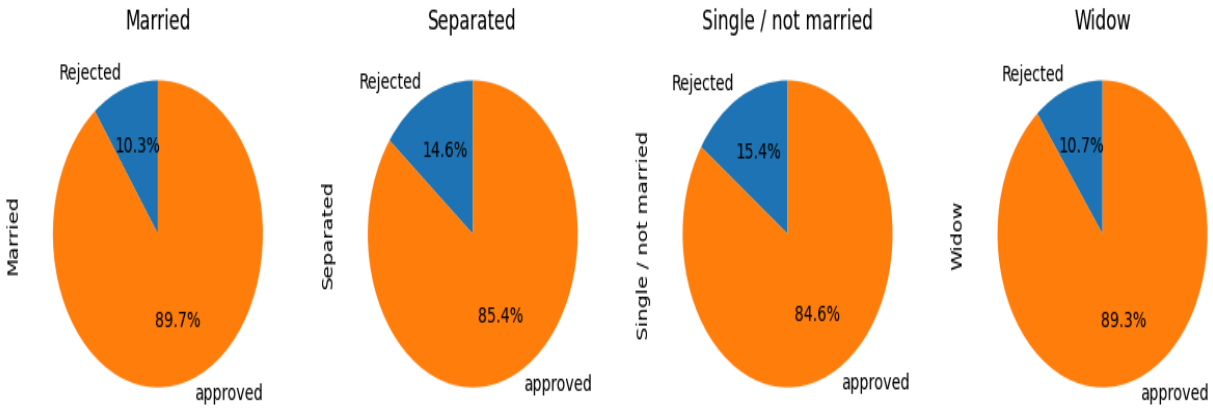

Annual income vs Application status
Annual_income
400000
300000
200000
100000
0
Rejected
approved
App_status

Insights from violin plot

The medean annual income is higher for approved applicants than for rejected

this suggests that approved applicants are generally wealthier than rejected application
Hence we may establish a relation between income and approval of credit cards

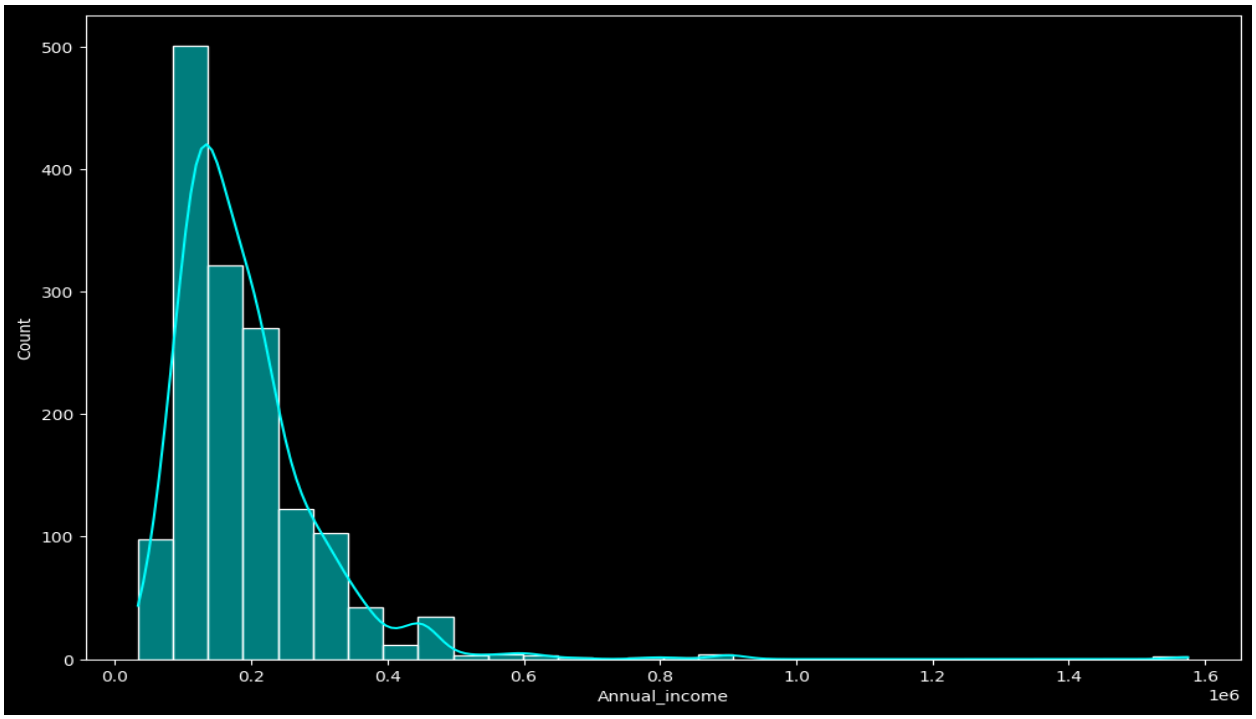
So these kind of insights for companies could use to more targeted lending programs



There are no much difference in rejection rate for different marital status and rejections are in the range of 10 to 16% only

---

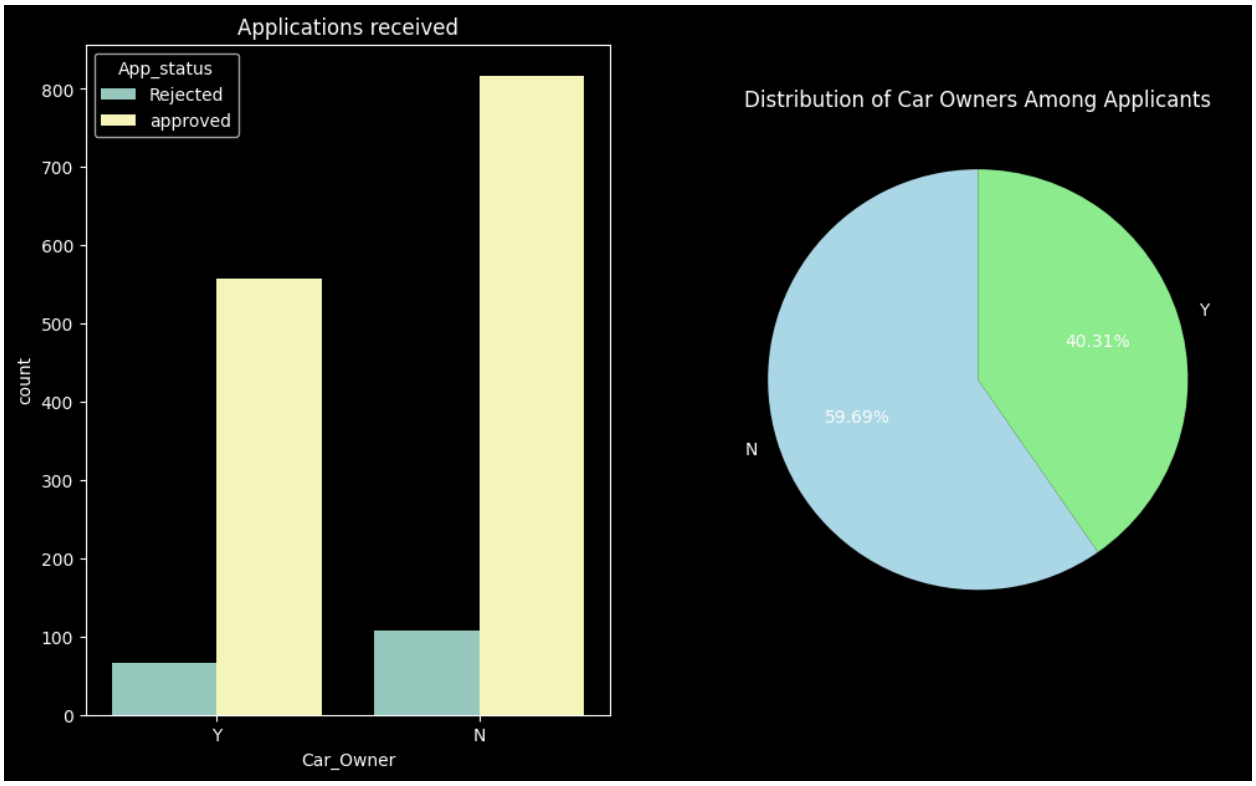
Disrtibution of annual income :



Distribution of annual income is skewed towards right and having wide range of income and also many outliers are present

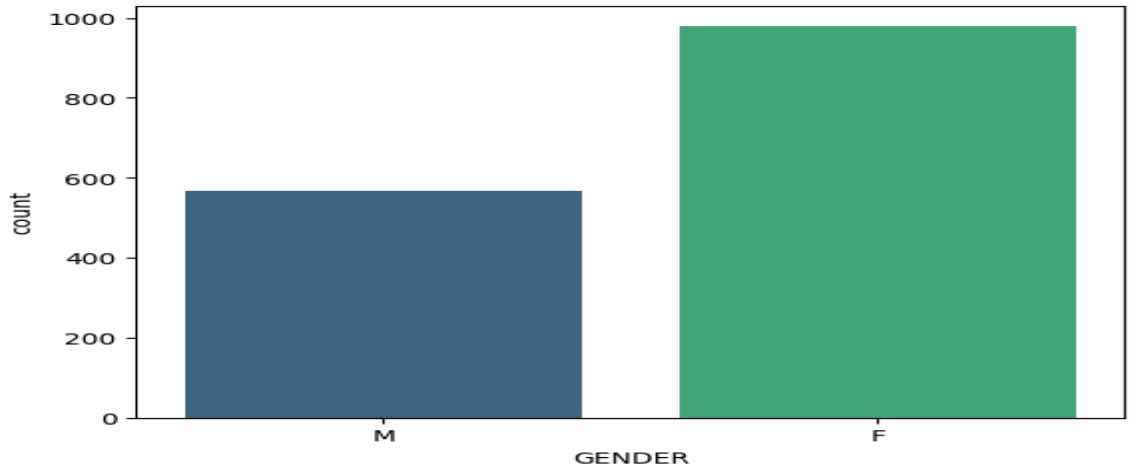
Highest income is 16 lakhs
Lowest is 33K

PIE Chart and bar graph car-owners and application status



"The percentage of credit card applicants who own a car is 40.31%, while non-car owners have a high approval rate. This suggests that car ownership may not be a significant factor influencing credit card approval.

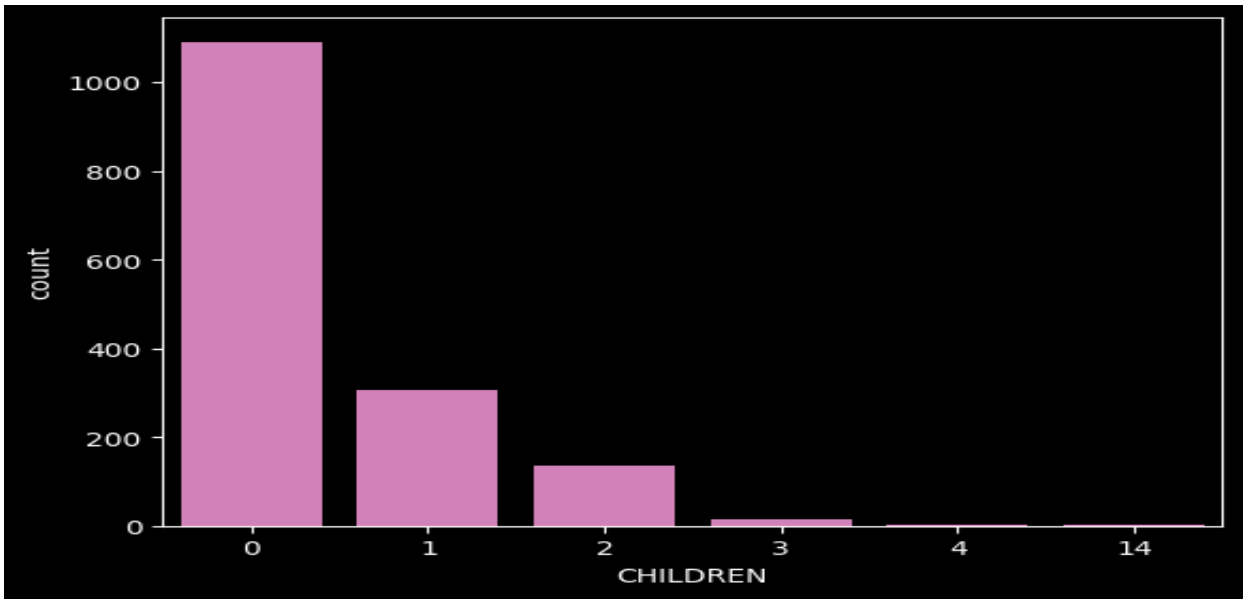
"GENDER VS APPLICATION STATUS:



More number of applications are came from Female compared with Male according to this data , male lending seekers are 50% lesser than female

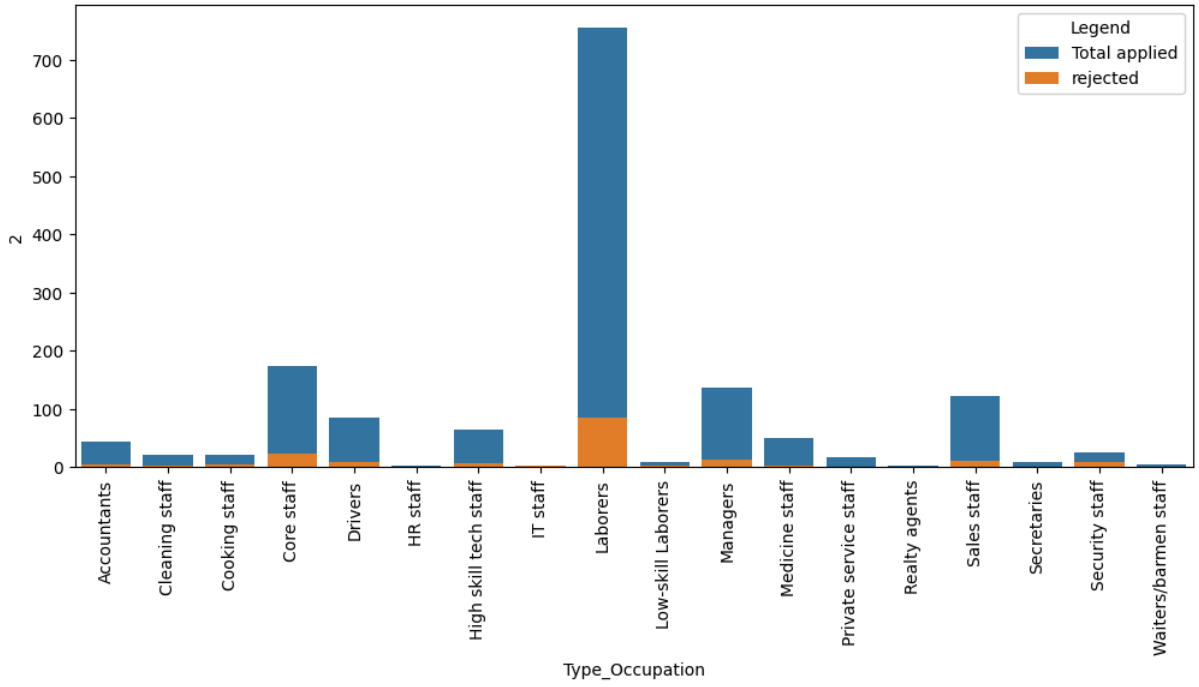
---

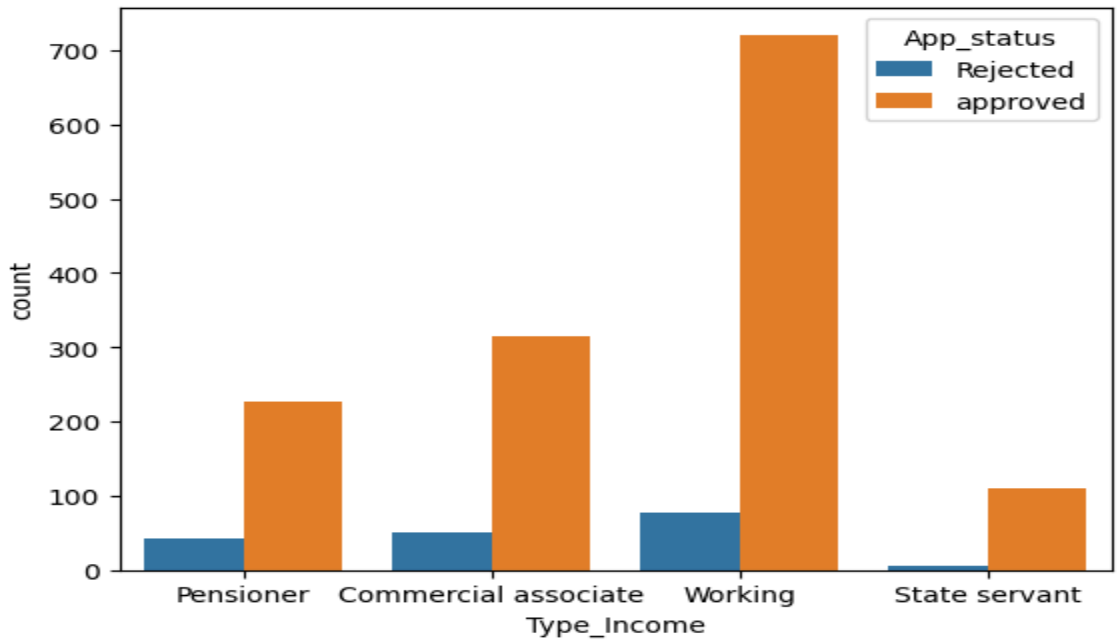
From below two graphs family with two numbers and zero children having more number of applications and they are approvedGraph with children vs applications count:



Couples with no children are more having bad credits

Bar graph between type occupation and credits





Above two graphs are saying working labour are having good credit and income earning from from state servant are having high good credits

---

3.2.5.Feature Engineering:

Identify relevant features and engineer new ones to enhance the model's predictive power.

Feature creation: here we provided with some variables having birthday count and working days and these things are converted into age and working experience in years So these kind of creations are used for data analysis

Feature selection:

Column Redundancy and Removal:

Upon identifying a strong correlation between the presence of children and family members, the redundant feature is eliminated. Similarly, newly created columns like

age and working experience hint at redundancy with existing columns such as birthday count and working days count.

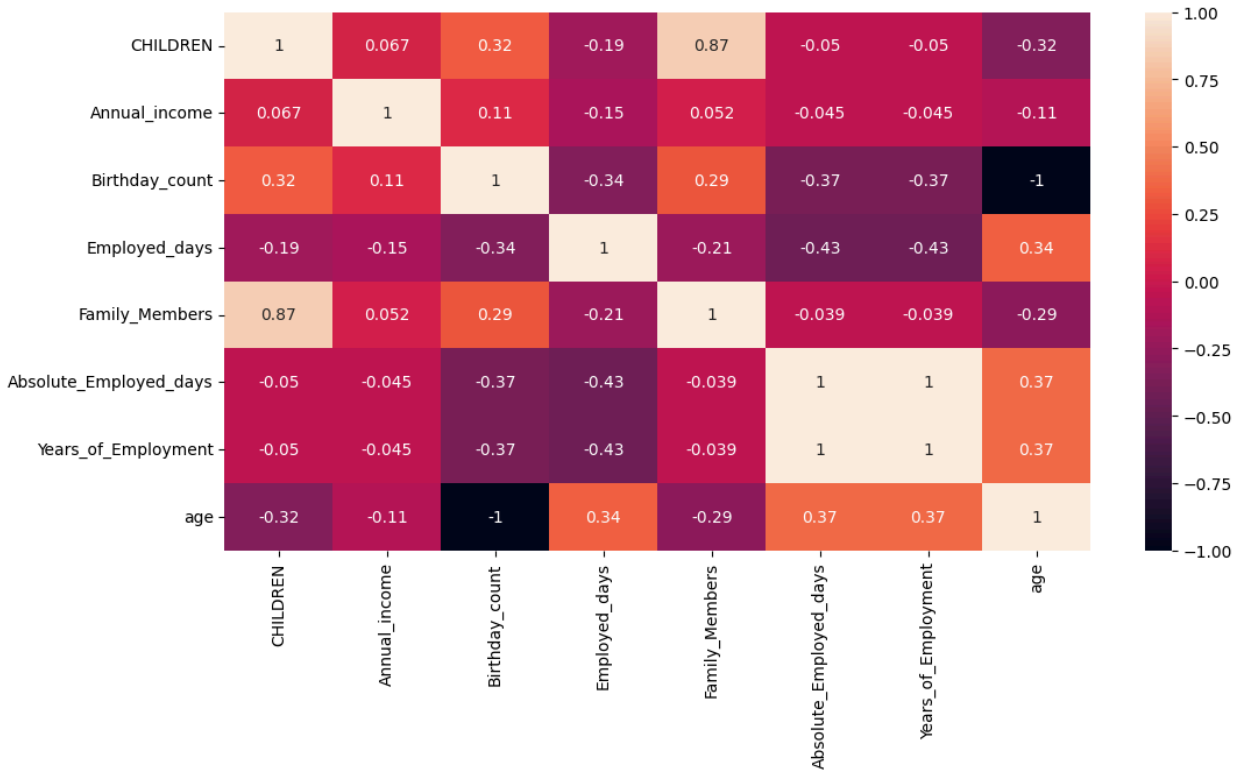
Features lacking predictive importance, including ID columns, and relationships like work-phone being linked to having a phone are recognized. Moreover, the universal ownership of mobile phones renders the mobile column non-contributory to prediction; hence, these columns are discarded.

While certain columns lack individual statistical significance with the target variable, they are retained. This decision is rooted in the understanding that, in combination with other variables, these columns may exert influence on the label. Techniques like Wrapping and embedded methods in feature selection account for variable interactions, making them instrumental in retaining potentially impactful features.

Objective:

Provide a streamlined rationale for the removal of redundant and non-contributory columns, highlighting the importance of considering interactions for comprehensive feature selection.

A correlation heat is shown following



Children is much correlated with family members ,age with birthday count,employee days with employe years some are created so removed before model training

3.2.6.FEATURE ENCODING

We have target variable which is categorical hence it is classification model and there is a need for transforming categorical data into numerical

Encoding Categorical Variables:

Convert categorical variables into a numerical format that machine learning models can understand. Common methods include one-hot encoding, label encoding, or target encoding.

Target Variable Encoding:

The target variable, having binary outcomes, was mapped using 0 and 1.

The mapping facilitated a clear distinction between approved (0) and rejected (1) credit card applications.

Independent Variable Encoding:

For remaining independent variables, a two-step approach was implemented.

Binary categorical variables were encoded using 0 and 1, creating separate columns for each category.

The get_dummies function was employed for nominal variables, generating distinct columns for each category and assigning binary values

Checking data balanced or not:

Check for data balanced or not as it is a classification model this effects on choosing metrics

3.2.7.FEATURE SCALING

Scaling and Normalization:

Standardize or normalize numerical features to a similar scale. This ensures that features with larger magnitudes do not dominate the learning process.

Standardization (Z-score normalization):

When to Use:

Standardization is often used when features in the dataset have different units or scales.

It is suitable for algorithms that assume a Gaussian distribution of the input features, such as linear regression, logistic regression, or support vector machines.

Why:

It centers the data around zero and scales it by the standard deviation.

Helps algorithms converge faster during training.

Maintains the shape of the original distribution and does not bound values within a specific range.

Normalization (Min-Max scaling):

When to Use:

Normalization is suitable when features have varying ranges and you want to scale them to a specific range, typically [0, 1].

It is commonly used in algorithms that involve distance calculations, such as k-nearest neighbors or clustering algorithms.

Why:

Scales the data to a specific range, preventing features with larger magnitudes from dominating.

Maintains the relative relationships between data points.

Here i used standardization as this model is not a distance calculation and also having larger magnitudes in annual income ,age and employment years

3.2.8.DATA SPLITTING

Splitted into two datasets one is target variable and other one is independent variables
Divide the dataset into training and testing sets. The training set is used to train the model, while the testing set assesses its performance on unseen data.

Here initially train_set is 80% of data and other 20% for the testing of the data using python scikitlearn library So there will be xtrain xtest ytrain and ytest

3.2.9.Model selection :

Hypothesis Testing Approach:

Initiated with logistic regression as the baseline, proceeded to evaluate Decision Tree, and explored ensemble techniques (Random Forest, XGBoost) to assess model performances and validate the null hypothesis.

3.2.10.Model Training :

The machine learning algorithm was implemented using Python's scikit-learn library. The dataset, previously split into 80%, was utilized for training the model.

3.2.11.Model testing :

The machine learning algorithm was implemented using Python's scikit-learn library. The dataset, previously split into 20%, was utilized for testing the model.

3.2.12.Model Evaluation:

Machine learning model evaluation is a critical step in assessing the performance and effectiveness of algorithms. Metrics play a pivotal role in quantifying how well a model generalizes to new, unseen data. This guide provides an overview of key metrics used for classification and regression tasks, outlining their definitions, calculations, and interpretation.

3.2.13.Classification Metrics:

Accuracy

Purpose: Measures the overall correctness of predictions.

Interpretation: High accuracy indicates a well-performing model but may not be suitable for imbalanced datasets.

Precision

Purpose: Measures the accuracy of positive predictions.

Interpretation: Useful when minimizing false positives is crucial.

Recall (Sensitivity or True Positive Rate)

Purpose: Measures the ability to capture all positive instances.

Interpretation: Important when minimizing false negatives is critical.

F1 Score

Purpose: Balances precision and recall.

Interpretation: Suitable for imbalanced datasets, offering a compromise between precision and recall.

Confusion Matrix

Purpose: Illustrates model performance by comparing actual and predicted classifications.

Components: True Positives, True Negatives, False Positives, False Negatives.

Receiver Operating Characteristic (ROC) Curve

Definition: A graphical representation of the trade-off between true positive rate and false positive rate at various thresholds.

Components: Area Under the ROC Curve (AUC-ROC) quantifies the model's ability to distinguish between classes.

3.2.14.Model Development:

Train the XGBoost RFE model to predict creditworthiness, giving importance to recall and AUC-ROC score.

Feature engineering:

It is a iterative process as we do until satisfied results appear, wrapper method RFE is used with xgboost in project

RFE helps in identifying and selecting the most relevant features for a given problem, potentially improving model performance by reducing noise and overfitting.

It selected the following features

Annual_income', 'Family_Members', 'Years_of_Employment', 'age',
'Phone_1', 'GENDER_M', 'Car_Owner_Y', 'Housing_type_House / apartment',
'Housing_type_Municipal apartment', 'Type_Income_Pensioner',
'Type_Income_State servant', 'Type_Income_Working',
'Marital_status_Single / not married', 'Type_Occupation_Drivers',
'Type_Occupation_Laborers', 'Type_Occupation_Managers',
'Type_Occupation_Sales staff', 'Type_Occupation_Security staff'

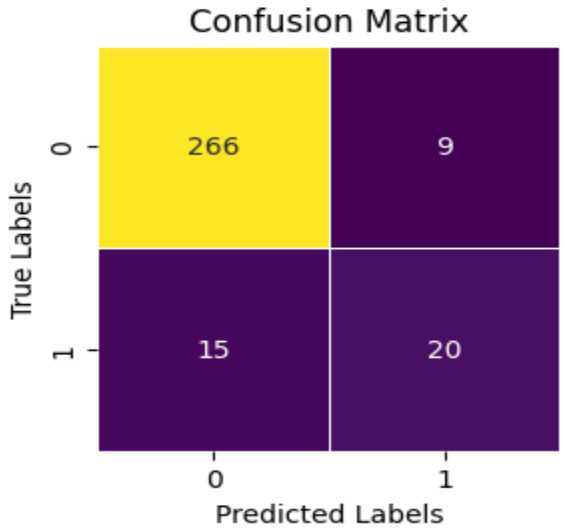
Metrics:-xgboost RFE

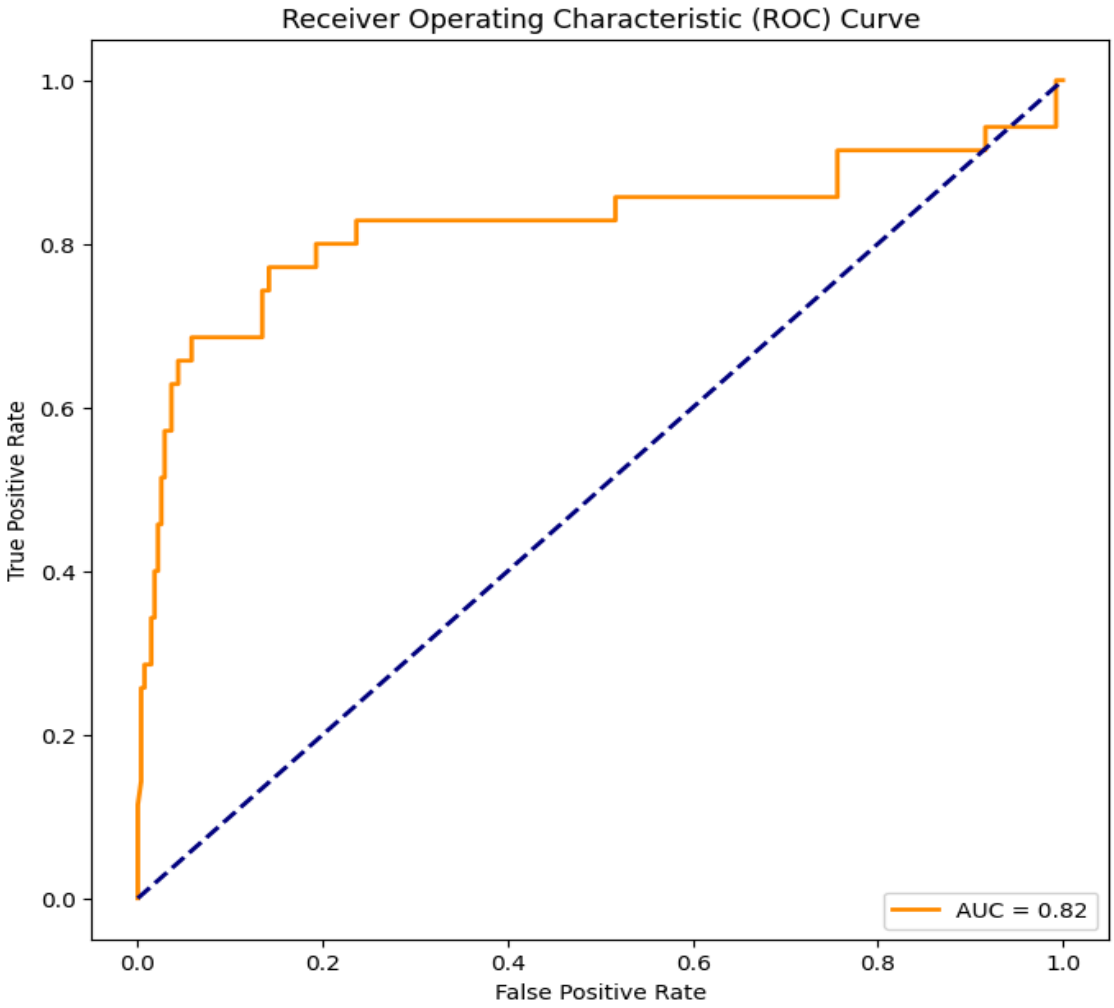
precision_score 0.6896551724137931

recall_score 0.5714285714285714

accuracy_score 0.9225806451612903

f1_score 0.625


Confusion Matrix
266
9
15
20
0
1
True Labels
0
1
Predicted Labels


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 0.82

3.2.15.Cross validation :

Cross-validation is a resampling technique used in machine learning to assess the performance of a model and mitigate the risk of overfitting. It involves dividing the dataset into multiple subsets, training the model on different combinations of these subsets, and then evaluating its performance. The primary goal is to obtain a more robust estimate of the model's performance by testing it on different subsets of the data.

Stratified K-Fold Cross-Validation:

In classification tasks, this variation ensures that each fold maintains the same class distribution as the entire dataset.

Precision-recall curve:

creating a Precision-Recall Curve and then applying a custom threshold to classify predictions.

Purpose: Assess the trade-off between precision and recall across different probability thresholds.

Procedure: Predicted probabilities (y_prob) are used to generate precision and recall values for varying thresholds.

Plot: The Precision-Recall curve visually represents the model's performance across different decision thresholds.

Custom Threshold Application:

Objective: Optimize the model's trade-off between precision and recall based on specific requirements.

Adjustment: A custom threshold (custom_threshold = 0.2) is chosen from the curve.

Prediction: Model predictions are adjusted using the custom threshold to classify instances.

Threshold Tuning: The custom threshold is adjusted based on the Precision-Recall curve, influencing the model's classification behavior.

In summary, the Precision-Recall Curve helps visualize the precision-recall trade-off, and the custom threshold allows fine-tuning the model's classification based on specific priorities, balancing precision and recall according to the application's requirements.

metrics:-

precision_score 0.7083333333333334

recall_score 0.4857142857142857

accuracy_score 0.9193548387096774

f1_score 0.576271186440678

3.2.16.Over sampling:

Oversampling is a technique used to address class imbalance in a dataset by increasing the number of instances in the minority class. When one class is significantly underrepresented compared to the others, machine learning models may have difficulty learning patterns in the minority class. Oversampling aims to mitigate this issue by creating a more balanced distribution of classes.

SMOTE (Synthetic Minority Over-sampling Technique):

Generates synthetic instances for the minority class by interpolating between existing instances.

Creates new examples by considering the feature space between existing instances rather than simply replicating them.

Helps prevent overfitting by introducing variability.

metrics (validation set):-

precision_score 0.14035087719298245

recall_score 0.24242424242424243

accuracy_score 0.7016129032258065

f1_score 0.17777777777777776

3.2.17.HYPERPARAMETER TUNING:

Optimizing a machine learning model involves tuning hyperparameters, external configurations crucial to a model's behavior. Follow these steps:

Selection of Hyperparameters: Identify key hyperparameters like learning rate or regularization strength.

Define the Search Space: Specify valid ranges for each hyperparameter.

Choose a Search Strategy:

Grid Search: Exhaustively evaluate all hyperparameter combinations within the defined space.

Evaluation Metric: Define a metric (e.g., accuracy, precision) to assess each hyperparameter's impact.

Cross-Validation: Utilize cross-validation to gauge model performance across varied training subsets.

Hyperparameter Tuning Process:

Train the model on the training data for each hyperparameter combination.

Evaluate on the validation set using the chosen metric.

Repeat until optimal hyperparameters are identified.

Select Best Hyperparameters: Determine the combination yielding the best validation set performance.

Final Model Evaluation: Assess the final model's performance on an independent test set for robust generalization.

Results:

metrics (validation set):-

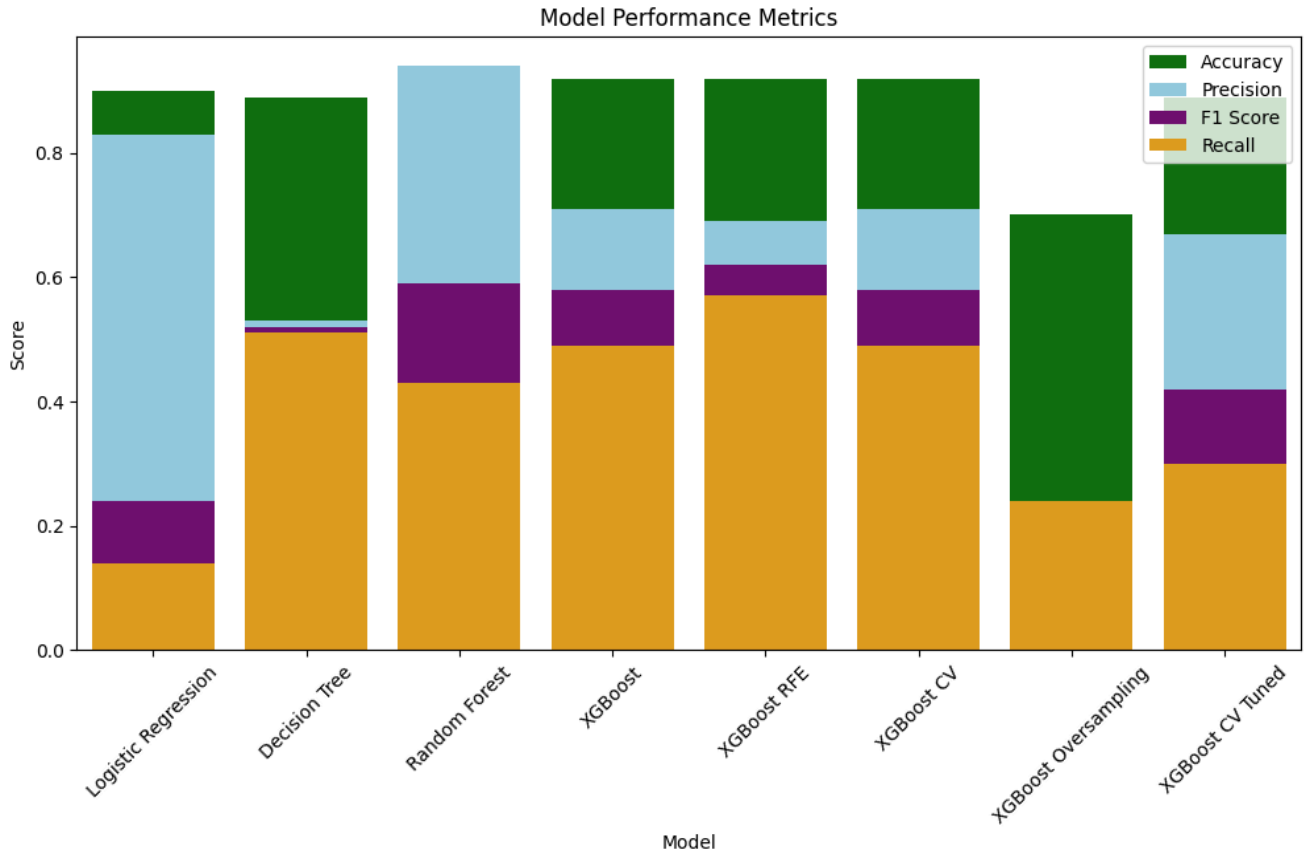
precision_score 0.6666666666666666

recall_score 0.30303030303030304

accuracy_score 0.8870967741935484

f1_score 0.41666666666666663

3.2.18.Comparison of all Algorithms:



Model Recommendations for Credit Card Approval:

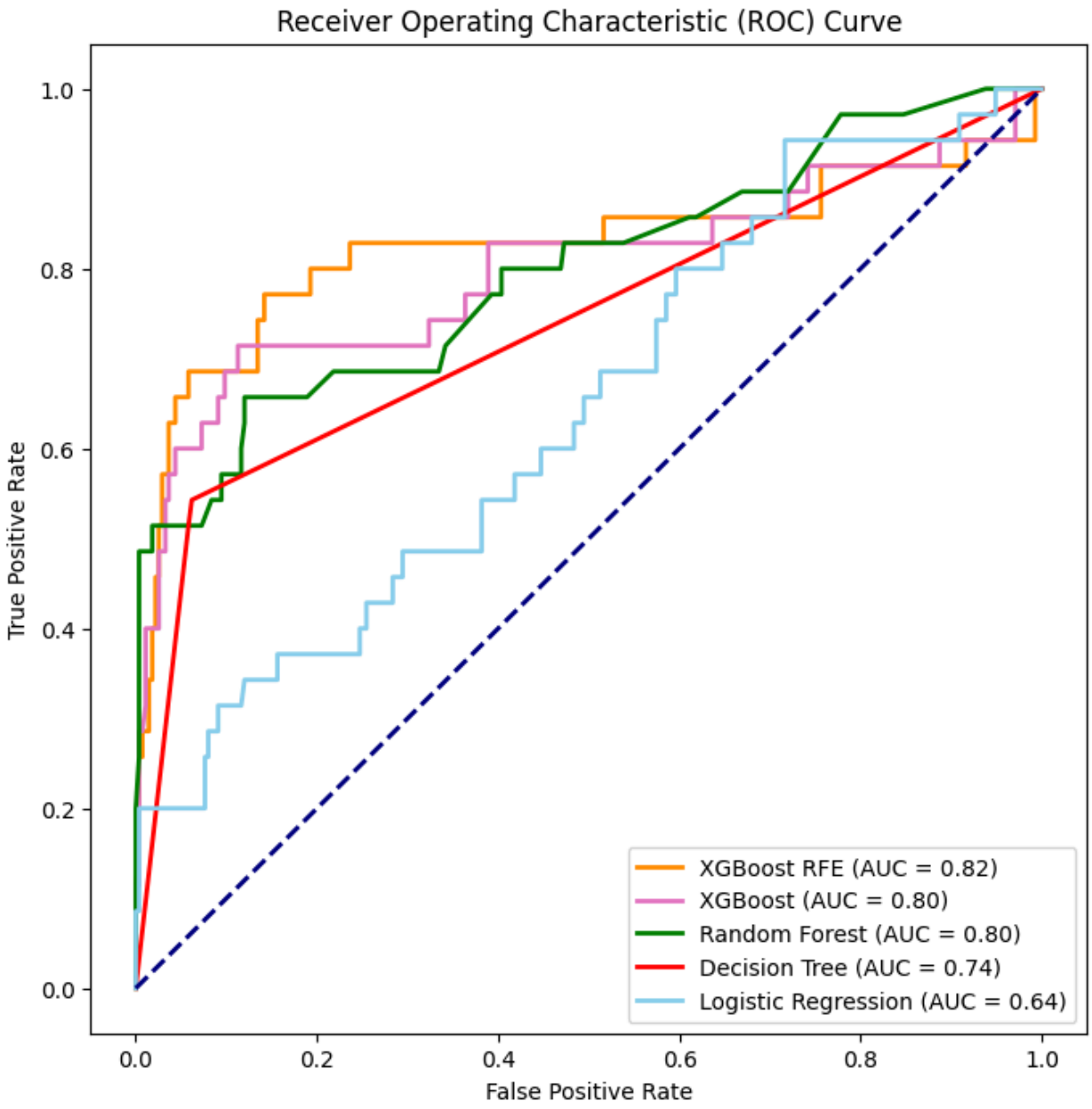
Considering recall, AUC score, and balanced precision:

XGBoost RFE (Recursive Feature Elimination):

Advantages: Demonstrates the best overall performance, making it a strong candidate for credit card approval.

Strengths: High true positive rate is crucial for correctly identifying creditworthy applicants.

AUC: The highest AUC score of 0.82 indicates effective discrimination between creditworthy and non-creditworthy individuals.



XGBoost:

Performance: Close to XGBoost RFE, suggesting robustness and effectiveness for credit scoring**.**

AUC: A commendable AUC score of 0.80 enhances its suitability for credit card approval.

Random Forest:

Competitive: Performs well and is suitable for ensemble learning in credit risk assessment.

Consideration: Ensemble methods, like Random Forest, can provide stability and accuracy in predicting creditworthiness.

Decision Tree:

Performance: Decent, but may be surpassed by ensemble methods in credit risk modeling.

AUC: Reasonable AUC score of 0.74, indicating moderate discriminatory power.

Logistic Regression:

Limitation: Exhibits the lowest discriminatory power among the models.

Consideration: May be suitable for simpler credit scoring scenarios but might lack the complexity needed for accurate risk assessment.

More Recommendations for Credit Card Approval:

Priority on True Positives: Given the context of credit card approval, prioritizing true positives (correctly approving creditworthy applicants) is crucial to minimize risk.

Ensemble Methods: XGBoost RFE, XGBoost, and Random Forest are preferred for their ensemble learning capabilities, providing robust and accurate predictions.

Model Transparency: Consider the interpretability of the selected model, especially if explaining credit decisions to customers is a regulatory requirement.

Feature Importance Analysis: Conduct a detailed analysis of feature importance to understand the factors influencing credit approval decisions.

Considering Metrics:

After analyzing the performance metrics of different models for credit card approval:

Model	Precision	Recall	Accuracy	F1 Score
Logistic Regression	0.83	0.14	0.9	0.24
Decision Tree	0.53	0.51	0.89	0.52
Random Forest	0.94	0.43	0.93	0.59
XGBoost	0.71	0.49	0.92	0.58
XGBoost RFE	0.69	0.57	0.92	0.62
XGBoost CV	0.71	0.49	0.92	0.58
XGBoost Oversampling	0.14	0.24	0.7	0.18
XGBoost CV Tuned	0.67	0.3	0.89	0.42

Precision: Indicates the accuracy of positive predictions. Crucial in credit card approval to avoid approving risky applications.

Recall: Crucial for identifying all positive instances. In credit card approval, high recall ensures that potentially creditworthy applicants are not overlooked.

Accuracy: Measures the correctness of predictions. Important to strike a balance between precision and recall.

F1 Score: The harmonic mean of precision and recall, providing a balance between the two. Useful in scenarios with imbalanced classes.

XGBoost RFE:

Balance: Good balance between precision (0.69) and recall (0.57).

F1 Score: High F1 score (0.62).

Recommendation: Strong contender for credit card approval, providing a good trade-off between false positives and false negatives.

Random Forest:

Performance: High precision (0.94) and reasonable recall (0.43).

F1 Score: Achieves a good balance reflected in the F1 score (0.59).

Recommendation: Performs well and can be considered, especially for ensemble learning.

XGBoost CV Tuned:

Balance: Balanced performance with precision (0.67) and recall (0.30).

F1 Score: F1 score of 0.42.

Consideration: Suitable if interpretability is a concern, and further fine-tuning might enhance performance.

XGBoost and XGBoost CV:

Consistency: Demonstrate consistent performance with reasonable precision, recall, and F1 scores.

Consideration: Can be considered based on specific requirements and preferences.

XGBoost Oversampling:

Challenges: Low precision (0.14) and recall (0.24).

F1 Score: Lower F1 score (0.18).

Recommendation: Suggests challenges in distinguishing between positive and negative instances.

Recommendations:

***Model Selection*: XGBoost RFE is recommended for its balanced precision and recall, suitable for credit card approval with a focus on minimizing both false positives and false negatives.**

***Ensemble Methods*: Random Forest and XGBoost show competitive performance and can be considered, especially if ensemble methods are favored.**

Identified gaps in knowledge and incorporating relevant features can significantly contribute to improving the predictive power of the model. Here are some key features that, when included, may enhance the model's performance:-

Credit History: Incorporate: Include details about the applicant's credit history, encompassing previous loans, repayments, and credit scores.

Debt-to-Income Ratio: Calculate the ratio of existing debt to annual income to gauge an applicant's financial obligations and capacity for additional credit.

Credit Utilization:

inclusion: For individuals with credit history, incorporate the percentage of available credit currently in use. High utilization may indicate financial strain.

Income Stability:

Evaluation: Assess the consistency and stability of annual income over a specified period. Sudden fluctuations could be noteworthy.

Financial Reserves:

Inclusion: If available, add information about an individual's savings or financial reserves. This provides insights into their ability to handle financial shocks.

4. Conclusion

This project aims to revolutionize the credit card approval system by introducing advanced data science techniques, focusing on recall and AUC-ROC score, and incorporating dynamic adaptability. The proposed solution aligns with the evolving needs of the credit card department, offering a robust and innovative approach to creditworthiness assessment.

This sample proposal provides a framework for a credit card approval system project, outlining the problem statement, project overview, proposed solution and deliverables .

After comprehensive analysis and evaluation of various machine learning models for credit card approval, the XGBoost with Recursive Feature Elimination (XGBoost RFE) emerges as the optimal choice. This model demonstrates a superior recall of 0.57, highlighting its effectiveness in correctly identifying creditworthy applicants, a critical factor in credit card approval scenarios.

Key Metrics for XGBoost RFE considering credit card approval

***Recall**: 0.57*

***AUC-ROC Score:** High, contributing to the model's ability to discriminate between positive and negative instances.*

The emphasis on recall underscores the significance of minimizing false negatives, ensuring that deserving applicants are not incorrectly rejected. The associated AUC-ROC score reinforces the robustness of the model in capturing the trade-off between true positive rate and false positive rate.

##Hypothesis Testing Results :-

In the context of the null hypothesis, which posits that logistic regression is better than any other machine learning algorithms for credit card approval, the analysis rejects this null hypothesis. The empirical evidence from the model evaluations, particularly in favor of XGBoost RFE, contradicts the superiority of logistic regression.

Rejection of Null Hypothesis:

Reasoning*: The superior recall and AUC-ROC performance of XGBoost RFE indicate that more complex and adaptive models can outperform logistic regression in credit card approval scenarios.*

Final Decision:

Based on the presented evidence, XGBoost RFE stands out as the preferred model, providing a well-balanced and effective solution for credit card approval, especially when prioritizing recall and discrimination power.

5.Tools Used:

- Google Collab
- Python libraries
- Google Documentation

6.Deliverables

XGBoost RFE Model: A trained machine learning model optimized for credit card approval..python ipynb file

Documentation: Comprehensive documentation covering the data sources, model development, and system implementation. Pdf

Worked datasets two CSV files